

www.arduino.cc

LEARN TO LOVE ARDUINOS WITH iGEM!

Contents

The iGEM Foundation	2
How highschoools can get involved in iGEM	2
The Arduino Board	3
The Breadboard	3
How Arduinos Work in Digital Mode	4,5
Digital vs Analogue	6
Coding in C	7,8
Arduino Maths	9
Learn More About Arduinos	10
Activities for Arduinos	10



The Internationally Genetically Engineered Machine Foundation



Non-profit



Open Source



Synthetic Biology



International



Collaboration



Education



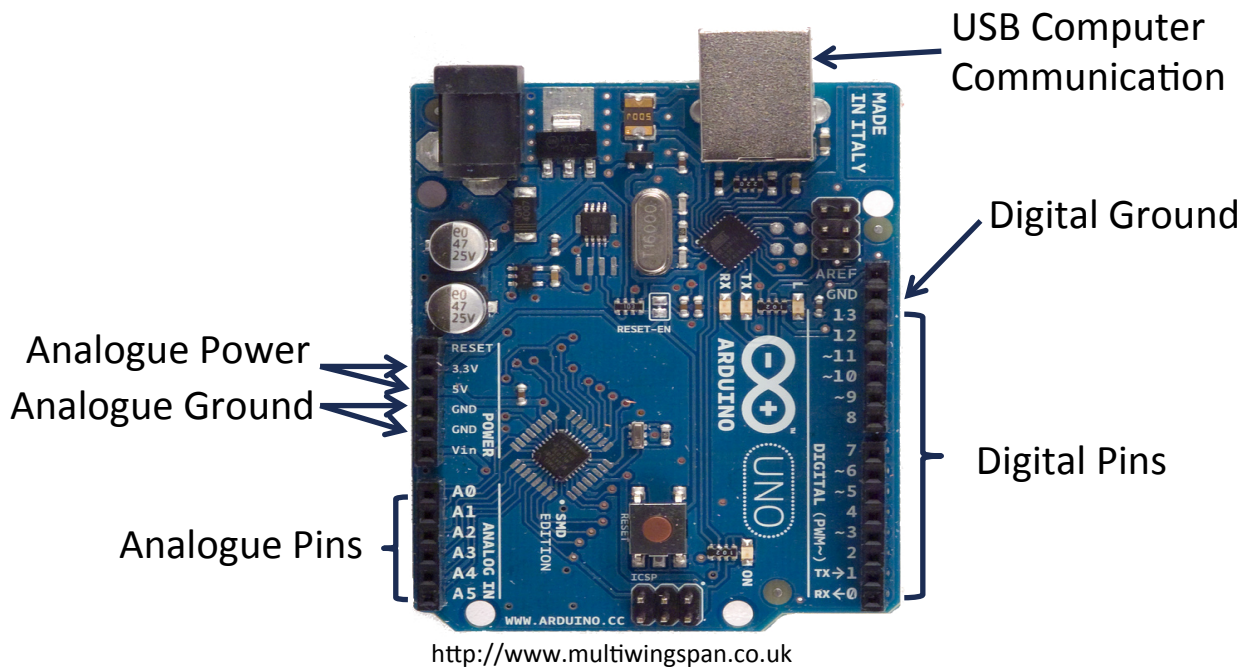
Competition

iGEM High School Division

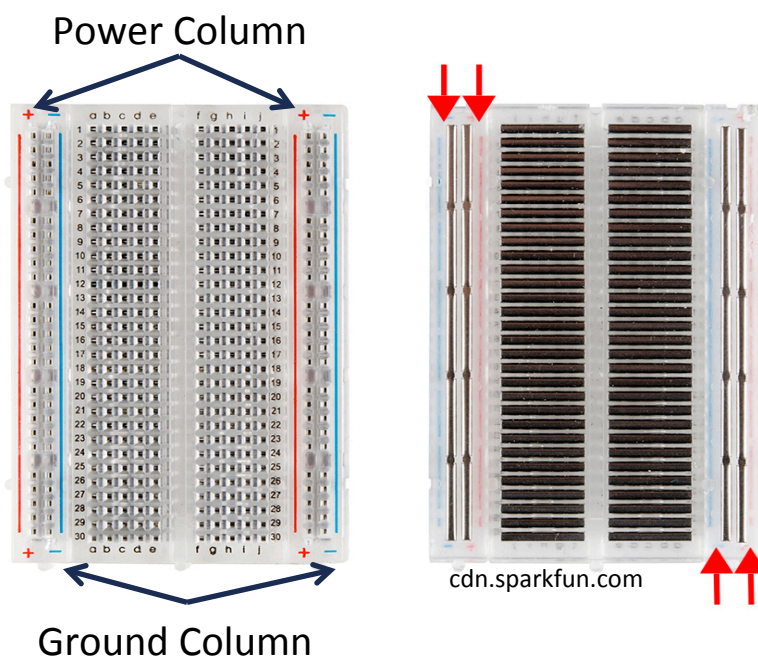
Highschools have been part of iGEM since 2011, with their own separate competition. As of this year high schools will have their own track in the main competition allowing them to compete, not only against other schools, but universities across the world as well. They are required to work on a project, make a wiki page and poster, and present in Boston just like all the other teams.

<http://2015.igem.org/Tracks/HighSchool>

Arduino Board



Breadboard



How the Arduino Works in Digital Mode

Digital Read: Tell if pin is on or off

Pin to test (from 0-13)

`int pinValue = digitalRead(6);`

Look at pin 6, tell me whether
it is **HIGH**
(someone is pushing a button)
or
LOW (button not being
pressed).
Save this value in the variable
“pinValue”.

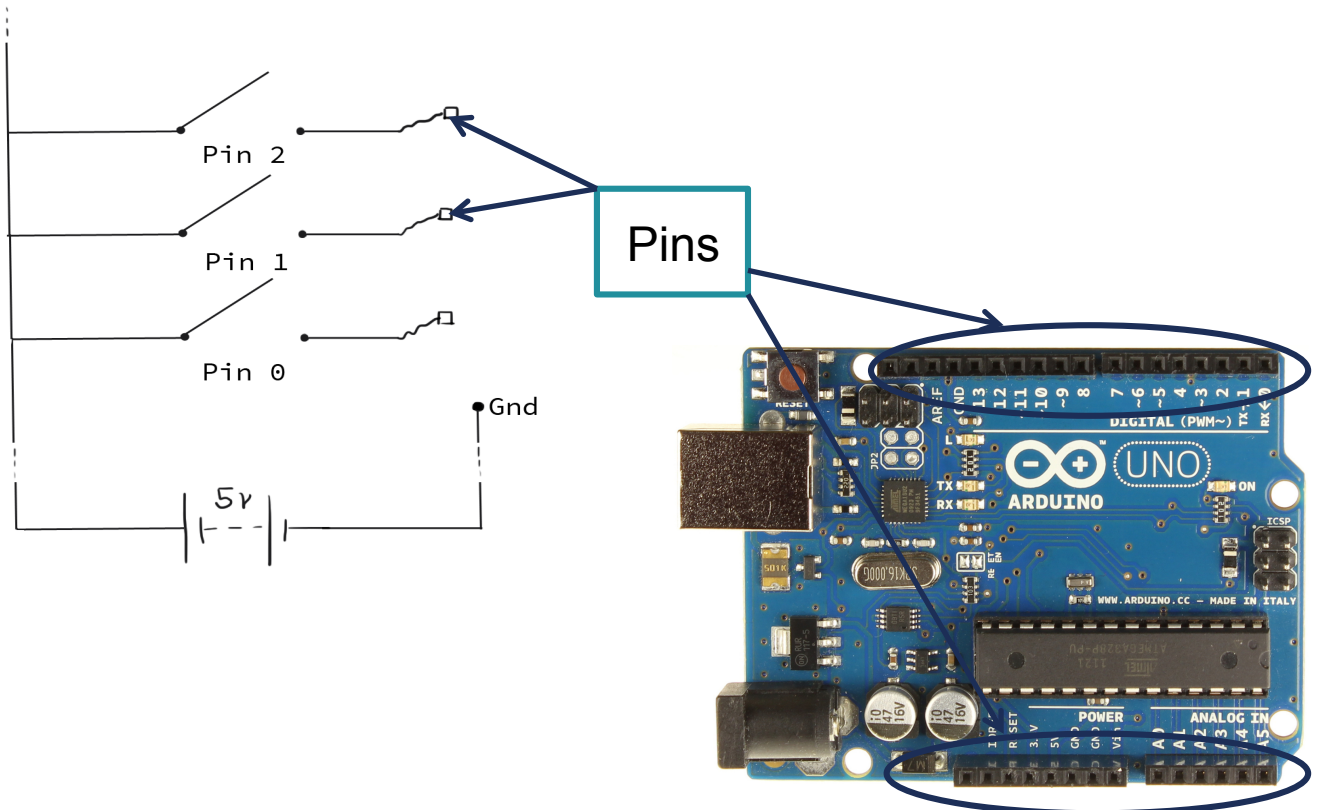
Digital Write: Turn pin on or off

Pin to set (from 0-13)

`digitalWrite(6, HIGH);`

Voltage to set it to – **HIGH** or
LOW

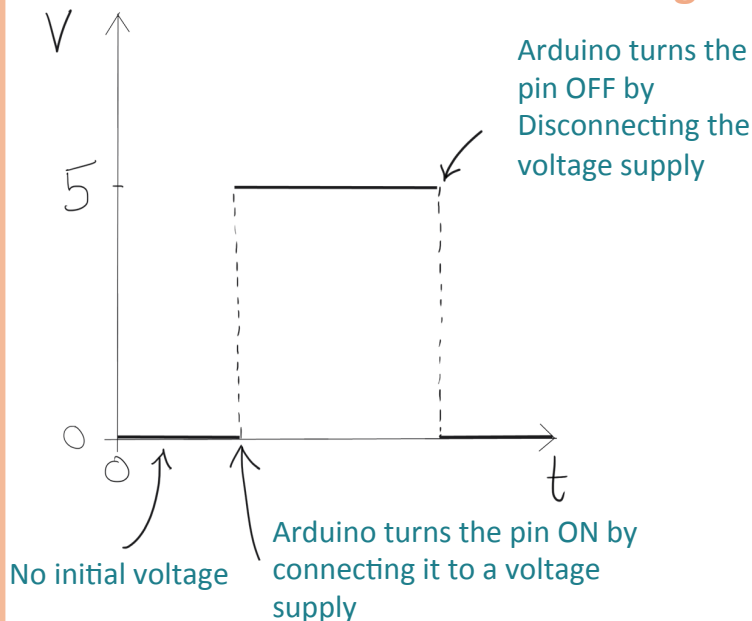
How the Arduino Works in Digital Mode



The pins are where you attach the circuits you make to the Arduino (e.g. an LED). These circuits are connected from the pin to the “ground” pin – you can just think of this as **connecting the circuit** to a little **power supply/battery**. The Arduino controls the current through this circuit, simply by opening or closing a switch. This switch is controlled by the program you load onto the Arduino. When you write “digitalWrite(3, HIGH)” you are actually telling the Arduino to close the switch next to pin 3 so the circuit connected to pin 3 is powered.

Digital vs Analogue Voltage

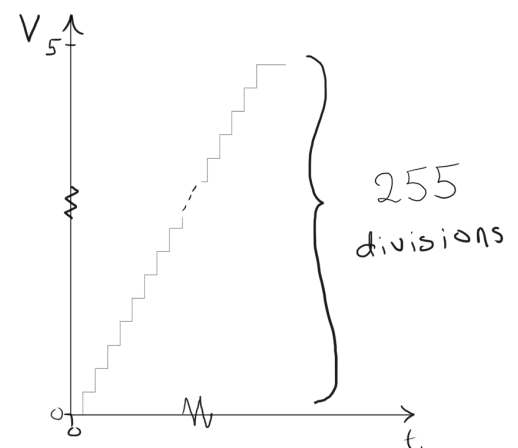
Digital



The only possible voltages are **HIGH** or **LOW** – only two possibilities. HIGH is full voltage (e.g. 5 volts) whereas low is off (0 volts).

Analogue

There are supposed to be a continuous range of voltages for “analogue” pins. Think of it like a dimmer switch on lights – you can vary the brightness of the lights by varying the voltage input, controlled by some sort of dial. The analogue pins are supposed to be like this – but they aren't.



They actually only take a **set range** of input voltages (in exactly the same as digital pins) except they can take 255 different voltage inputs (so, evenly spaced inputs from 0 to 5 V for example). This can be used to give the **illusion of an analogue** input.

Coding in C++

// (double slash): This, placed before any code, will not be read by the computer. It is a comment to let you or others know what that piece of code does.

//:** This carries out the same function as above, except allows for several lines of code. (Text goes between the **).

{ } (curly brackets): These follow a function being called and contain inside them all the code that is the information (statements) being fed to that function. If you use an opening curly bracket you must make sure to close it later in your code.

; **(semicolon):** Shows the point where a statement ends.

setup function():

- Called when a sketch starts.
- Runs only once, after a power up or reset.
- Function initialises the variables and pins.

```
void setup() {  
    /* put your setup code  
    here, to run once: */ }  
}
```

loop function():

- Runs a section of code repeatedly.
- The program can change and respond between loops.

```
void loop() {  
    /* put your main code  
    here, to run repeatedly: */ }  
}
```


For Loop

```
for (initialisation; termination condition; increment)
{
    statement(s)      }
```

If Statement – to implement logic

```
if (test criteria)
{
    //if the above test criteria is true, run the codes within the brackets.
    statement(s)      }
else if (another test criteria)
{
    statement(s)      }
else
{
    //This code is run if the two above test criteria are false.
    statement(s)      }
```

The language of computers: **Binary**

011011101110

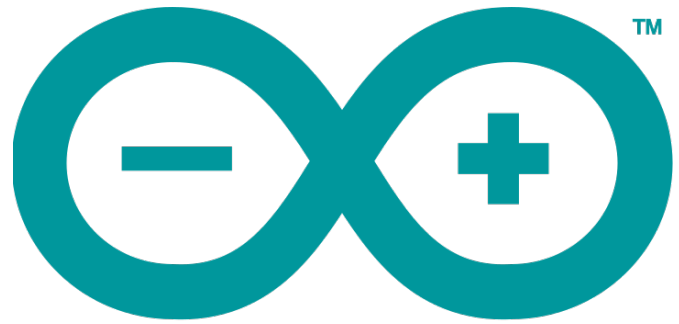
Bit-can be in one of
two states: ON (1) or
OFF (0)

Types: indicate how bits will be stored. These are important as tells the computer how to interpret the binary

- **int:** number
- **float:** decimal number
- **string:** series of characters
- **bool:** has one of two values: true (1) or false (0)

Arduino Maths

+ : addition
 - : subtraction
 = : assignment operator
 * : multiplication
 / : division
 % : modulo



<http://assets.hgbrasil.com>

"=" means something different

Normally, "=" is saying 'the thing on the left is equal to the thing on the right'. In C, "=" is used to SET the variable on the left (e.g. x) to a specific value.

<code>int x = 7;</code>	Creates a variable called x and sets it equal to 7.
<code>x = x + 3;</code>	Sets value of x to its previous value plus 3 (x now equals 10)
<code>x = x * 4;</code>	Sets value of x to its previous value times 4 (x now equals 40).

What happens?

The right of the equals sign is calculated first – so you can use your variable (above, this is x) in this calculation. After the calculation finishes, the value on the right is saved. The variable on the left of the equals sign is then set as this value. This is how "x = x + 3" makes sense.

Note:** You can **UPDATE** a variable simply by using the **variable name

Want to learn more about Arduinos?

The Arduino Website itself is a fantastic website with loads of teaching resources, projects to carry out with your arduino and an extensive help page, covering any problems you may have.

<https://www.arduino.cc>

Where to buy your own Arduino.

- <https://store.arduino.cc>
- <http://www.adafruit.com>
- <https://shop.pimoroni.com>

Some more great projects



10 Simple-But-Fun Projects to Make with Arduino
<http://www.popularmechanics.com/technology/how-to/a3099/10-simple-but-fun-projects-to-make-with-arduino-15603196/>

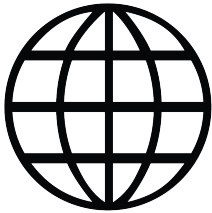


10 Great Arduino Projects for Beginners
<http://www.makeuseof.com/tag/10-great-arduino-projects-for-beginners/>



Arduino Project Ideas
<http://playground.arduino.cc/Projects/Ideas>

Please keep following the team to see how we get on in the competition!



<http://www.clipartbest.com>

<http://2015.igem.org/Team:Cambridge-JIC>



<http://freeiconbox.com>

<https://www.facebook.com/CambridgeJIC2015>



<http://animprobablefiction.com>

<https://instagram.com/iGEMCambridge>



<http://cdn.flaticon.com>

<https://twitter.com/iGEMCambridge>